CMSC733 Homework 0 - Alohomora!
Using 2 Late Days

Nishad Kulkarni
Master of Engineering in Robotics
University of Maryland, College Park
Email: nkulkar2@umd.edu

***Abstract*—This document serves as the report for the first homework assignment for CMSC733 Spring 2022 Class submission for the student Nishad Kulkarni. Phase 1 was chosen for the submission.**

I. Introduction

Computer vision deals with many subsets, one of which is boundary/edge detection. This homework aims to analyse the Probability of Boundary detection algorithm. This algorithm performs better than the traditional Sobel and Canny baselines. For this algorithm three filter, namely, Derivative of Gaussian (DoG), Leung-Malik, and Gabor Filters are used. These filter banks were implemented in the code.

II. Filter Banks

We now look at the filters that were implemented in this assignment. Given below is a short discussion for the same.

A. Derivative of Gaussian Filters

Derivative of Gaussian, as the name suggests, is the first derivative of a 2 dimensional Gaussian. A simple method to create this filter is to convolve a 2D Gaussian with the Sobel filter. This is given by:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{1}$$

or, by:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2}$$

The equation for the 2D Gaussian is given by:

$$G(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{(x-\mu_x)^2+(y-\mu_y)^2}{2\sigma_x\sigma_y}} \tag{3}$$

For the DoG filter, the values sigmas are assumed to be equal and the centers are also assumed to be equal.

In this work, a DoG filter bank with 2 scales and 16 orientations was created. This is what it looks like.

The original DoG filter at the extreme left were rotated according to the angles based on chosen scale.

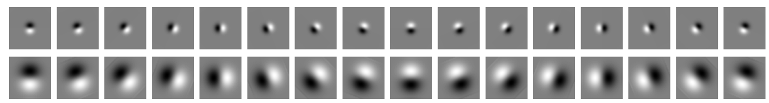

Fig. 1. DoG Filter Bank

B. Leung-Malik Filters

The Leung-Malik filter bank is made of 3 scales and 6 orientations of 1st order and 2nd order Gaussian derivatives. Also there are 8 Laplacian of Gaussians and 4 Gaussians. These total up to 48 filters in the filter bank.

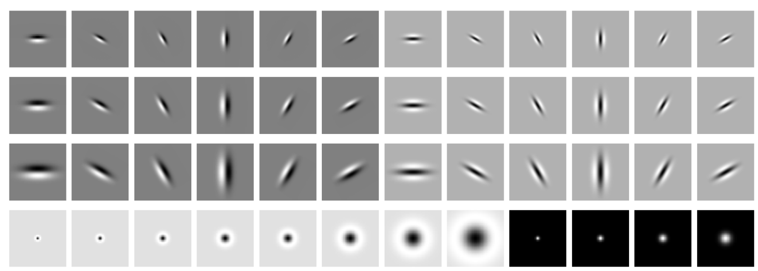

Fig. 2. Leung-Malik Filter Bank

C. Gabor Filters

The Gabor Filter is a 2D Gaussian mosdulated with a sinusoidal wave. The generated filter is shown below.

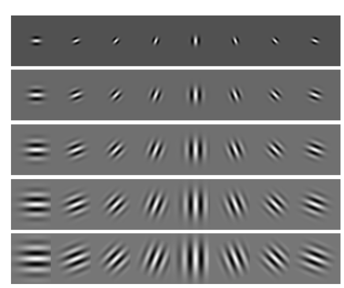

Fig. 3. Gabor Filter Bank

For this assignment, as can be seen above, Gabor filters of 5 scales and 8 orientations were generated.

III. Filtering

For proceeding to the goal the filters (in any desired combination) are convolved with the images. The resulting convolution is called a Texton Map. Along with this Texton map we also analyse the brightness and colour maps. Thes maps are denoted by T, B, and C respectively.

In this work, the DoG and Leung Malik-Filter Banks generated previously are utilized. After this, half disk maps are generated as shown below.

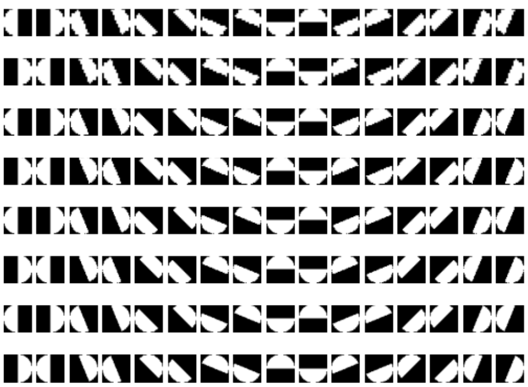

Fig. 4. Half Disk Masks

The gradients, denoted by T_G, B_G, and C_G are the gradients of T, B, and C maps. The resulting maps are shown below.

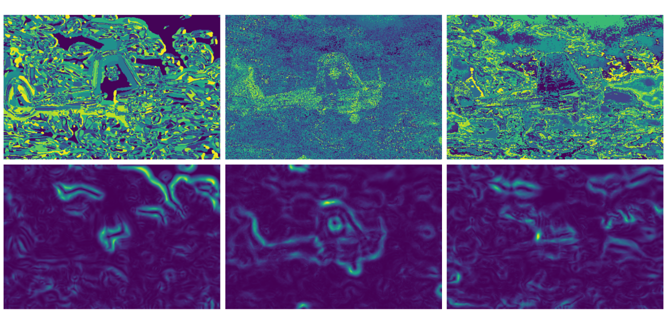

Fig. 5. T,B,C and T_G, B_G, and C_G of image 1

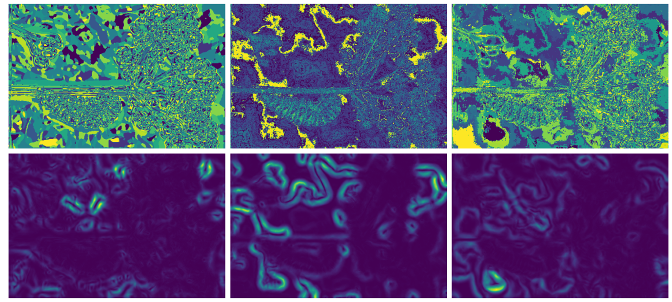

Fig. 6. T,B,C and T_G, B_G, and C_G of image 2

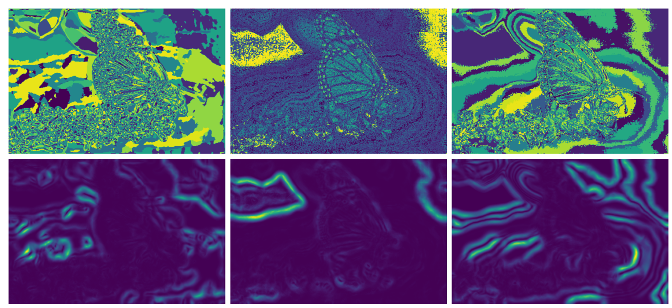

Fig. 7. T,B,C and T_G, B_G, and C_G of image 3

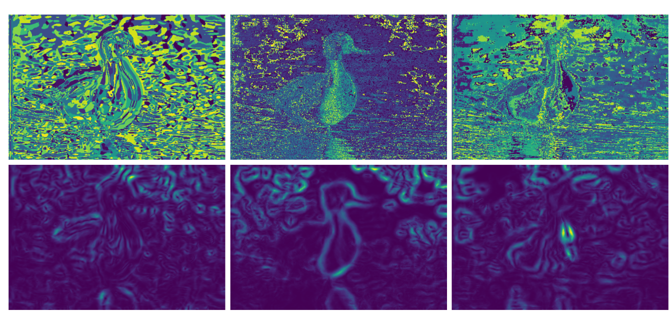

Fig. 8. T,B,C and T_G, B_G, and C_G of image 4

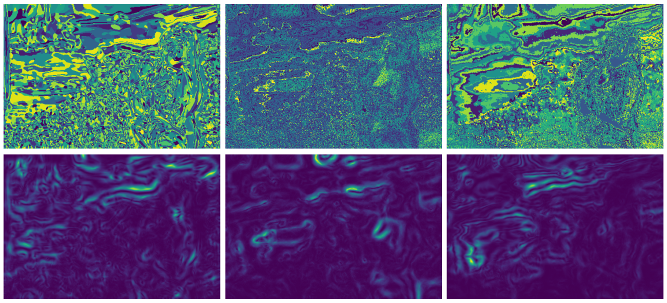

Fig. 9. T,B,C and T_G, B_G, and C_G of image 5

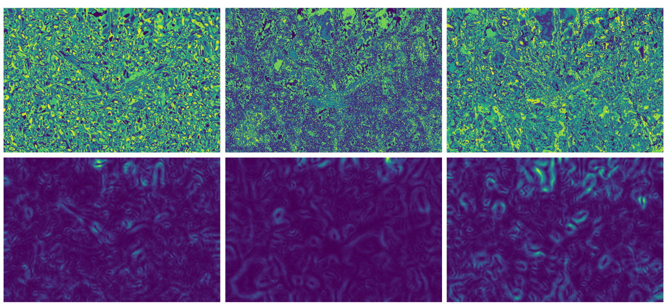

Fig. 10. T,B,C and T_G, B_G, and C_G of image 6

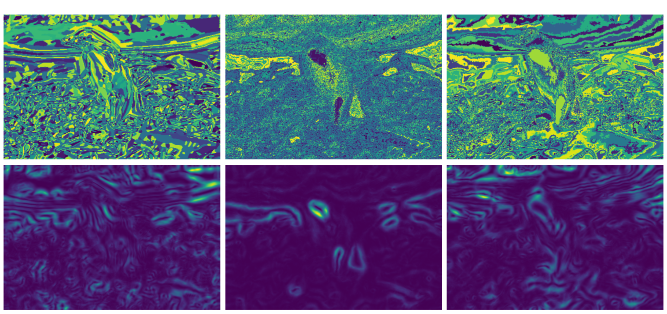

Fig. 11. T,B,C and T_G, B_G, and C_G of image 7

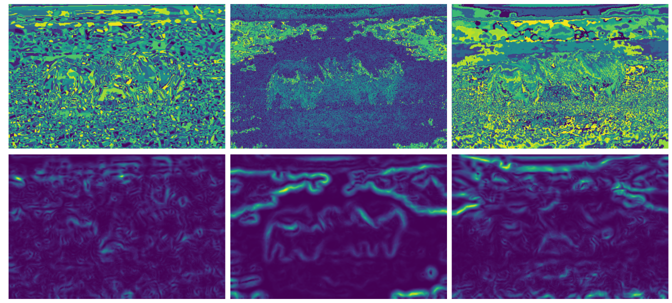

Fig. 12. T,B,C and T_G, B_G, and C_G of image 8

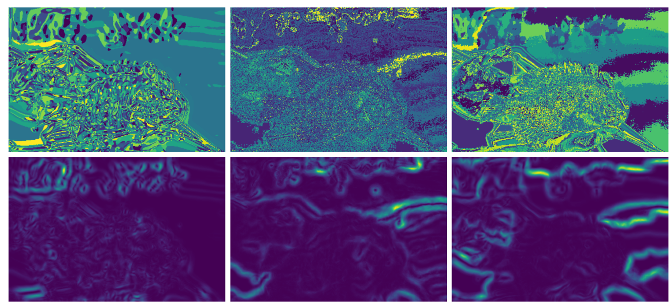

Fig. 13. T,B,C and T_G, B_G, and C_G of image 9

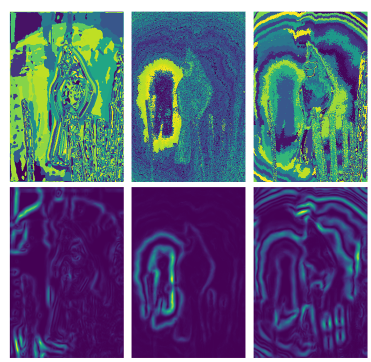

Fig. 14. T,B,C and T_G, B_G, and C_G of image 10

IV. Pb-Lite Output

We finally reach our goal by averaging the Gradient Maps T_G, B_G, and C_G and then adding them to the weighted Sobel and Canny Baselines. The Final results are as shown below.

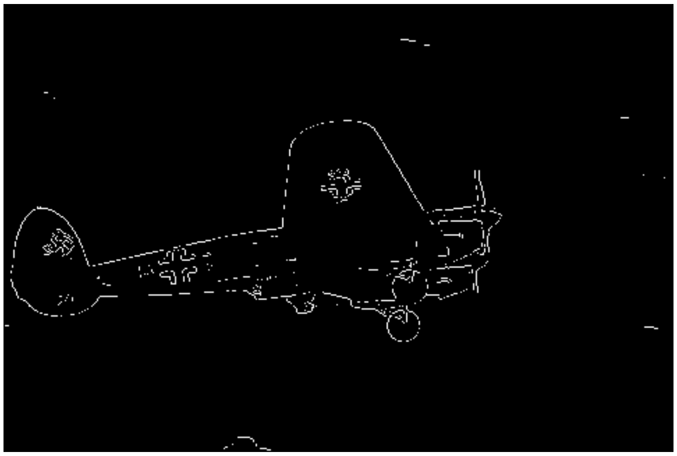

Fig. 15. PB-Lite Output of image 1

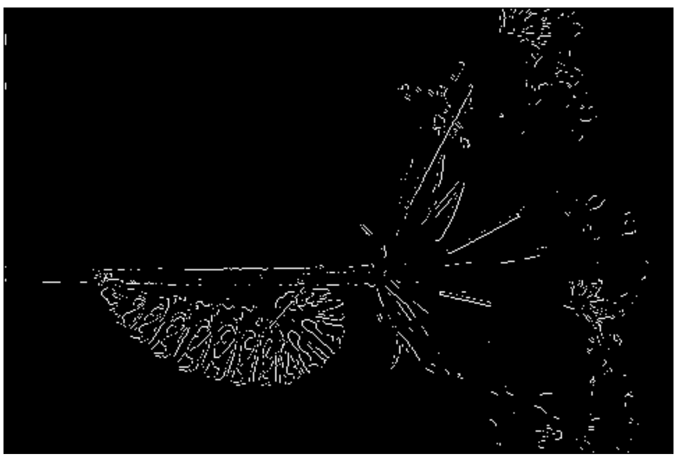

Fig. 16. PB-Lite Output of image 2

Fig. 17. PB-Lite Output of image 3

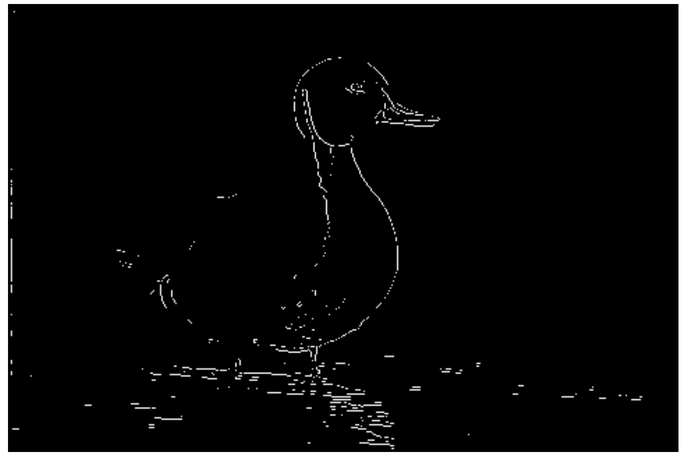

Fig. 18. PB-Lite Output of image 4

Fig. 19. PB-Lite Output of image 5

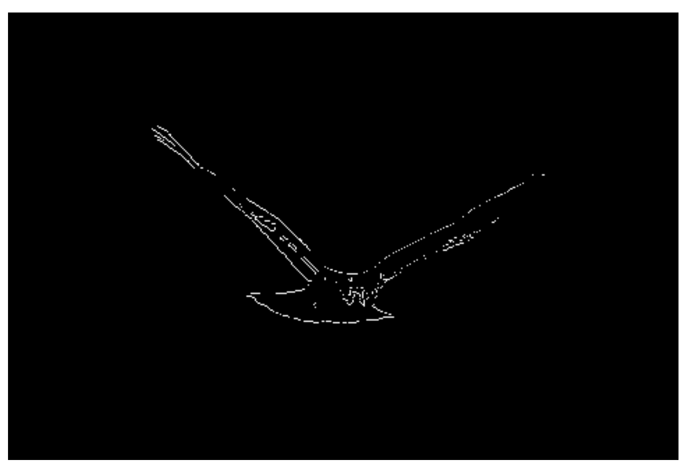

Fig. 20. PB-Lite Output of image 6

Fig. 21. PB-Lite Output of image 7

Fig. 22. PB-Lite Output of image 8

Fig. 23. PB-Lite Output of image 9

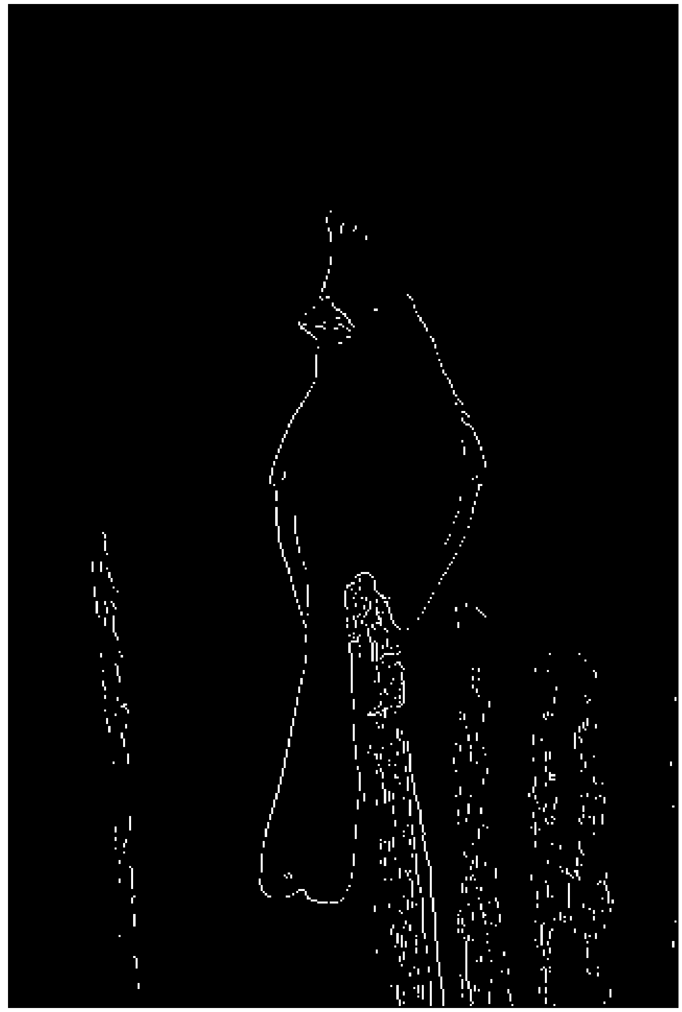

Fig. 24. PB-Lite Output of image 10

V. Conclusion

As seen above, the output for PB-Lite in this assignment is not extremely impressive except for the fact that it does manage to capture minute edges. The reason for this might be ineffecient aplication of gradients over T,B, and C for the generation of T_G, B_G, and C_G. The Textons, Bightness Maps, Colour Maps were run at a clustering number of 256 each, giving satisfactory results as shown above. The gradients of these were run for 16 as well as 64 iterations each, with no significant changes in results. I hope to improve on this work in the future.

Acknowledgment

I would like to thank his friends and batchmates Naveen Mangla and Arunava Basu, who helped point out errors and supported me throughout the project. I would also like to thank the anonymous posters on Piazza.

References

[1] Spring 2022, CMSC733 Course github repository and previous students' submissions.